

数值分析第一次上机练习实验报告

——解线性方程组与误差分析

力 4 杨昊光 2014011619

一、问题的描述

设 $H_n = [h_{ij}] \in \mathbb{R}^{n \times n}$ 是 Hilbert 矩阵, 即

$$h_{ij} = \frac{1}{i+j-1}.$$

对 $n = 2, 3, 4, \dots$,

(a) 取 $x = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n$, 及 $b_n = H_n x$. 再用 Gauss 消去法和 Cholesky 分解法来求解 $H_n y =$

b_n , 分析误差情况.

(b) 计算条件数 $\text{cond}(H_n)_2$.

(c) 使用某种正则化方法改善(a)中的结果.

(d) 用 SOR 迭代法和共轭梯度法求解 $H_n x = b_n$, 并与前面的直接方法作比较.

二、方法描述

1. Gauss 消去法: 若 $H_n y = b_n$, 则 $y = H_n^{-1} b_n$. 作增广矩阵 $[H_n \ b_n]$, 对每一行作顺序消去得上三角矩阵, 即 $[H_n \ b_n] \rightarrow \dots \rightarrow L_{nk}^{-1} [H_n^{(k)} \ b_n^{(k)}] \rightarrow \dots \rightarrow [U \ b'_n]$, 再进行回代解算出各 y . 其中,

$$L_{nk}^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\frac{2k-1}{2k} & 1 & \\ & & \vdots & & \ddots \\ & & -\frac{2k-1}{k+n-1} & & & 1 \end{bmatrix}$$

在 Matlab 中, Gauss 消去法默认是作列主元消去的, 即在顺序消去的每一步中选取

列主元, 用行交换矩阵左乘增广矩阵得 $I_{ik,k} [L_{n'}^{(k)} \ b^{(k)}]$, 使得每一次消去使用的是

第 k 到 n 行中最大的元素, 这样可以减小舍入误差.

2. Cholesky 分解法: 对对称正定矩阵 H_n , 存在唯一的对角元素为正数的下三角矩阵 L_n , 使得 $H_n = L_n L_n^T$. 其中对 L_n 有

$$\begin{cases} l_{11} = 1 \\ l_{ij} = \frac{1}{l_{jj}} \left(\frac{1}{i+j-1} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right), \quad j = 1, 2, 3, \dots, n; i = j+1, \dots, n. \\ l_{jj} = \left(\frac{1}{2j-1} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{\frac{1}{2}} \end{cases}$$

再依次求解 $L_n \tilde{y} = b_n$ 和 $L_n^T y = \tilde{y}$. 由于计算 L_n 的对角线元素存在较大舍入误差, 该

法对于双精度计算到 12 阶以后就不再精确，甚至会产生无法进行 Cholesky 分解的情况。

3. 条件数的计算: $\text{cond}(H_n)_2 = \|H_n\|_2 \|H_n^{-1}\|_2$ 。
4. Tikhonov 正则化方法: 求解 $H_n y = b_n$, 则当 α 充分小时可以将 $(\alpha I + H_n^* H_n) y = H_n^* b_n$ 的解作为原解的近似。由于 H_n 是实对称矩阵, $H^* = H^T = H^2$ 。由实验得 α 只需取一个很小的数值就可以大大减小 $\alpha I + H_n^* H_n$ 的条件数, 说明 H_n 的特征值都是很接近 0 的数 ($\sim O(10^{-13})$), 但是它们之间最大的和最小的又差了 10^n 数量级以上。
5. SOR 迭代法: 记 $y^{(k+1)} = B y^{(k)} + f$, 取

$$B = (D - \omega L)^{-1}[(1 - \omega)D + \omega U], \quad f = (D - \omega L)^{-1} \omega b_n.$$

其中, $H_n = D - L - U$, 取 $y^{(0)} = \mathbf{0}$, $\omega = 0.1$, 迭代 2000 次后查看结果。

6. 共轭梯度法: 迭代 100 次算法如下所示

(1) 取 $y^{(0)} = \mathbf{0}$

(2) 计算初始残差和下降方向 $r^{(0)} = b - H_n y^{(0)}$, $p^{(0)} = r^{(0)}$

(3) 对 $k=0, 1, \dots, 100$

计算下降距离	$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(p^{(k)}, H_n p^{(k)})},$
计算下降后位置	$y^{(k+1)} = y^{(k)} + \alpha_k p^{(k)},$
计算下降后的残差	$r^{(k+1)} = r^{(k)} - \alpha_k H_n p^{(k)},$
计算下降方向改正量	$\beta_k = \frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})},$
计算新的下降方向	$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}.$

三、 方案设计

通过编写 Matlab 程序生成 Hilbert 矩阵并进行上述运算。主程序文件为 `practice1.m`, 其中首先调用内部函数 `hilb(n)` 生成各阶 Hilbert 矩阵, 由于运算精度所限, 双精度数据只能最多生成 12 阶 Hilbert 矩阵进行 Cholesky 分解, 因此对相应的程序只取 $n=1$ 到 12, 其余的一直算到了 $n=30$ 。生成的矩阵存放到一个 cell 数组里, 按后续需要进行运算。同时生成对应的 b 矩阵, 同样存放到一个 cell 数组里。

Gauss 消去法为了尽量减少运算误差, 应避免数据的多次转存。这里调用 Matlab 自带的列主元 Gauss 消去法求解 $Ax=B$ 的运算符“\”, 同时计算误差为 $|y-x|$ 并储存。

Cholesky 分解法调用内部函数 `chol(H)` 进行, 计算得到 L , 将 L 和 L' 一同代入 y 的求解式进行运算, 同时计算误差并储存。由于 Cholesky 分解法对矩阵的正定性要求较高, 而当阶数较大时舍入误差对正定性判断及分解结果有较大影响, 此处对 Cholesky 分解法的解算只做到 $n=12$ 。

Tikhonov 正则化算法经计算得在大数加小数时小数的位数不丢失的前提下, $\alpha \sim O(\mu_1^2 \delta)$ 时得到的结果最好。经试验在这里我们取 $\alpha = 1 \times 10^{-12}$ 是可以获得最佳精度的 α 。

SOR 迭代法中, 我们采用迭代指定次数然后比较结果误差的形式, 迭代次数设为 1000 次。原计划对每个 n 求解最优松弛因子代入, 但是经过计算发现矩阵的接近奇异性令求特征值的舍入误差极大, 导致求得的松弛因子为复数, 因而最终舍弃了这一想法, 采用一个固定的 ω 。经过尝试取 $\omega=0.1$ 可以取得较好结果。

采用同样的方法对共轭梯度(CG)法进行分析, 发现其收敛速度相当的快, 最初迭代 1000 次结果较好, 于是降低迭代次数到 100 次, 结果仍然非常满意。

四、 计算结果及其分析

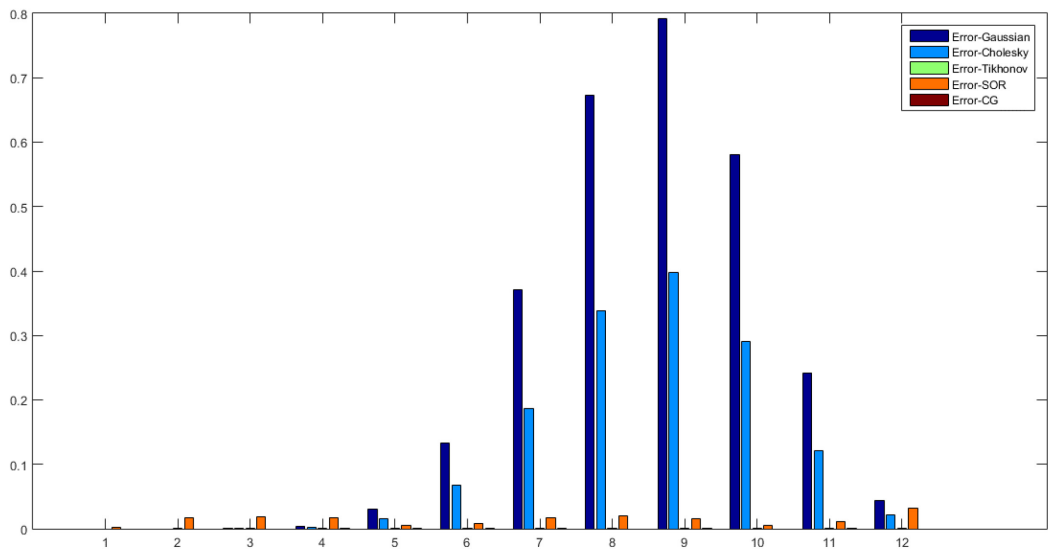
下面是我们根据程序计算结果得到的数据, 表中分别给出了 1~12 各阶的 2-条件数、采用不同方法得到的结果和误差值。从左到右各列分别为高斯消去法及其误差、Cholesky 分解

法及其误差、Tikhonov 正则化方法的误差、SOR 迭代法的误差和共轭梯度法的误差。

Gaussian	Error-G	Cholesky	Error-C	Error-Tikhonov	Error-SOR	Error-CG
n=1		cond(H ₁) ₂ =1				
1	0	1	0	1.0000889e-12	0.0000000E+00	0.0000000E+00
n=2		cond(H ₂) ₂ = 19.2814700679040				
1.000000000	4.441e-16	1.000000000	4.441e-16	4.3990145e-11	9.5368158E-14	0.0000000E+00
1.000000000	7.772e-16	1.000000000	7.772e-16	8.3981599e-11	1.6431301E-13	0.0000000E+00
n=3		cond(H ₃) ₂ = 524.056777586063				
1.000000000	9.992e-16	1.000000000	9.992e-16	1.7942520e-09	2.2385202E-03	1.7763568E-15
1.000000000	6.883e-15	1.000000000	6.883e-15	1.0134164e-08	1.2084221E-02	6.5503158E-15
1.000000000	7.438e-15	1.000000000	7.438e-15	9.8275188e-09	1.1478132E-02	4.8849813E-15
n=4		cond(H ₄) ₂ = 15513.7387389297				
1.000000000	1.499e-14	1.000000000	1.732e-14	6.9698037e-08	5.5671446E-04	4.8849813E-15
1.000000000	1.685e-13	1.000000000	1.923e-13	7.8956048e-07	4.7531130E-03	4.7184479E-14
1.000000000	4.130e-13	1.000000000	4.666e-13	1.9074433e-06	9.6278744E-03	1.0524914E-13
1.000000000	2.718e-13	1.000000000	3.051e-13	1.2429130e-06	5.5368864E-03	6.5947248E-14
n=5		cond(H ₅) ₂ = 476607.250242044				
1.000000000	1.610e-14	1.000000000	1.610e-14	2.3432150e-06	9.0258307E-04	3.4749981E-14
1.000000000	2.351e-13	1.000000000	1.277e-13	4.4492978e-05	8.1443896E-03	6.5458750E-13
1.000000000	8.544e-13	1.000000000	9.215e-14	1.9353833e-04	1.3222642E-02	2.7665648E-12
1.000000000	1.130e-12	1.000000000	3.447e-13	2.9391418e-04	1.9495918E-03	4.0951686E-12
1.000000000	4.970e-13	1.000000000	3.442e-13	1.4434523e-04	8.6182455E-03	1.9698687E-12
n=6		cond(H ₆) ₂ = 14951058.6406489				
1.000000000	7.413e-13	1.000000000	7.759e-13	5.4195575E-07	1.2730316E-03	8.1135099E-13
1.000000000	2.097e-11	1.000000000	2.191e-11	2.3577079E-05	1.0301361E-02	2.3150815E-11
1.000000000	1.411e-10	1.000000000	1.472e-10	1.9570234E-04	1.3976896E-02	1.5622925E-10
1.000000000	3.656e-10	1.000000000	3.811e-10	5.7616143E-04	4.6899507E-03	4.0551118E-10
1.000000000	4.026e-10	1.000000000	4.192e-10	6.9159127E-04	5.6461521E-03	4.4686033E-10
1.000000000	1.583e-10	1.000000000	1.648e-10	2.8959258E-04	4.9230795E-03	1.7584212E-10
n=7		cond(H ₇) ₂ = 475367356.896743				
1.000000000	8.459e-12	1.000000000	7.147e-12	3.7015683E-06	1.4047294E-03	1.3340393E-09
1.000000000	3.407e-10	1.000000000	2.864e-10	9.5006932E-05	1.1050615E-02	5.1468904E-08
0.999999997	3.305e-09	0.999999997	2.768e-09	5.4063117E-04	1.4725316E-02	4.8356985E-07
1.000000013	1.292e-08	1.000000011	1.079e-08	9.8191346E-04	5.6084592E-03	1.8437914E-06
0.999999976	2.378e-08	0.999999980	1.981e-08	1.7127129E-04	6.1958608E-03	3.3281276E-06
1.000000021	2.06e-08	1.000000017	1.714e-08	9.9649019E-04	8.0493210E-03	2.8395235E-06
0.999999993	6.79e-09	0.999999994	5.636e-09	6.3886260E-04	2.7813432E-03	9.2250710E-07
n=8		cond(H ₈) ₂ = 15257575589.2071				
1.000000000	6.739e-11	1.000000000	2.994e-11	4.6236473E-06	1.4157512E-03	1.4290158E-09
1.000000004	3.611e-09	1.000000002	1.604e-09	1.0371472E-04	1.1586959E-02	8.2036428E-08
0.999999953	4.714e-08	0.999999979	2.094e-08	4.9428025E-04	1.6147345E-02	1.1053574E-06
1.000000255	2.550e-07	1.000000113	1.133e-07	6.1863523E-04	7.2188329E-03	6.0993613E-06
0.999999313	6.865e-07	0.999999695	3.050e-07	3.0957743E-04	6.6679874E-03	1.6652174E-05
1.000000971	9.713e-07	1.000000431	4.315e-07	5.4649795E-04	1.2245601E-02	2.3820111E-05
0.999999309	6.912e-07	0.999999693	3.070e-07	5.7363970E-04	6.0490649E-03	1.7102879E-05
1.000000195	1.950e-07	1.000000087	8.662e-08	5.9186875E-04	1.1494975E-02	4.8616632E-06
n=9		cond(H ₉) ₂ = 493153733682.109				
1.000000000	4.132e-10	1.000000000	2.409e-10	2.7069699E-06	0.001458698	8.5992817E-09
1.000000028	2.819e-08	1.000000017	1.654e-08	4.9170846E-05	0.012617920	4.0355829E-07
0.999999527	4.733e-07	0.999999721	2.790e-07	1.5702437E-04	0.017697339	4.5989704E-06
1.000003358	3.358e-06	1.000001987	1.987e-06	1.3782876E-05	0.009949560	2.0843017E-05
0.999987743	1.226e-05	0.999992726	7.274e-06	2.7649708E-04	0.005373611	4.1675298E-05
1.000024932	2.493e-05	1.000014829	1.483e-05	3.4139812E-04	0.014599581	2.7038329E-05
0.999971456	2.854e-05	0.999982990	1.701e-05	4.8307858E-04	0.013504482	2.4767410E-05
1.000017196	1.720e-05	1.000010263	1.026e-05	6.1779642E-04	0.001817869	4.3693715E-05
0.99995761	4.239e-06	0.99997466	2.534e-06	6.1150853E-04	0.019217599	1.6931110E-05
n=10		cond(H ₁₀) ₂ = 16025170992026.3				
0.999999999	1.303e-09	0.999999999	1.177e-09	7.5107769E-07	0.001602847	3.5082313E-09
1.000000112	1.124e-07	1.000000101	1.009e-07	4.3919352E-07	0.014153552	1.4382890E-07
0.999997608	2.392e-06	0.999997864	2.136e-06	1.0528666E-04	0.018719190	2.8795150E-06
1.000021722	2.172e-05	1.000019326	1.933e-05	4.2465369E-04	0.013108270	1.3663971E-05

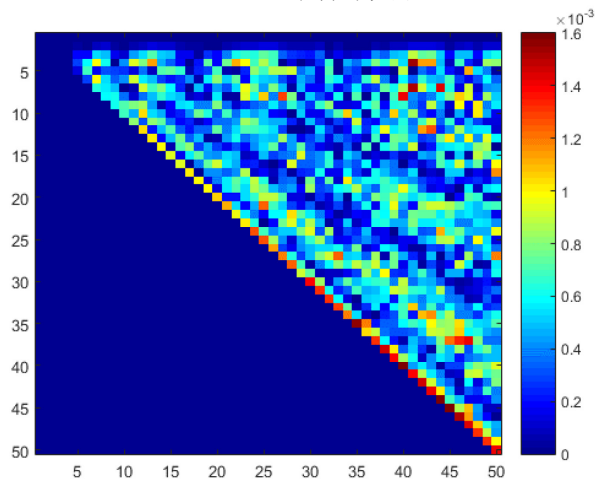
0.999896504	1.035e-04	0.999908188	9.181e-05	2.7498718E-04	0.002444122	2.0672060E-05
1.000284208	2.842e-04	1.000251545	2.515e-04	4.2058209E-04	0.014328910	2.9784802E-06
0.999534178	4.658e-04	0.999588471	4.115e-04	1.5228109E-05	0.017750486	2.4525042E-05
1.000449724	4.497e-04	1.000396714	3.967e-04	3.5756169E-04	0.011767406	5.3196371E-06
0.999764117	2.359e-04	0.999792177	2.078e-04	7.1986185E-04	0.002961719	3.2033213E-05
1.000051829	5.183e-05	1.000045617	4.562e-05	6.9009312E-04	0.025177507	1.4903403E-05
n=11		cond(H ₁₁) ₂ = 522153881275667				
0.999999999	1.101e-09	0.999999995	5.036e-09	1.7531760E-06	0.001834289	3.6800754E-09
1.000000105	1.053e-07	1.000000530	5.299e-07	6.1532418E-05	0.015850373	1.2484097E-07
0.999997473	2.527e-06	0.999986190	1.381e-05	4.3734132E-04	0.018910559	8.2313304E-07
1.000026374	2.637e-05	1.000155050	1.551e-04	9.2503673E-04	0.015884248	6.2584673E-07
0.999852283	1.477e-04	0.999072866	9.271e-04	1.3877813E-04	0.001243324	6.8141660E-06
1.000491189	4.912e-04	1.003270420	3.270e-03	9.2301630E-04	0.011984153	1.5490196E-05
0.998983468	1.017e-03	0.992858351	7.142e-03	7.1311407E-05	0.018748090	1.0382978E-06
1.001322795	1.323e-03	1.009761503	9.762e-03	1.2363091E-04	0.017584582	1.8014397E-05
0.998947363	1.053e-03	0.991872695	8.127e-03	7.9110966E-04	0.008612192	1.7494825E-06
1.000468059	4.681e-04	1.003768162	3.768e-03	4.6938404E-04	0.007374158	2.4375185E-05
0.999910893	8.911e-05	0.999254236	7.458e-04	8.0306702E-04	0.029328606	1.2331478E-05
n=12		cond(H ₁₂) ₂ = 1.63145371300616e+16				
0.999999927	7.280e-08	0.999999961	3.862e-08	3.3611221E-06	0.002101700	5.8970188E-09
1.000009333	9.333e-06	1.000004883	4.883e-06	9.2976384E-05	0.017342146	2.8742630E-07
0.999703805	2.962e-04	0.999846620	1.534e-04	5.2947796E-04	0.018300941	3.3450254E-06
1.004065245	4.065e-03	1.002088132	2.088e-03	7.8042553E-04	0.017775070	1.5230479E-05
0.970019454	2.998e-02	0.984699723	1.530e-02	3.5332397E-04	0.004827818	2.9387654E-05
1.132383826	1.324e-01	1.067205513	6.721e-02	5.6770880E-04	0.008518035	1.4835512E-05
0.629601135	3.704e-01	0.812783556	1.872e-01	9.1781923E-04	0.017301778	1.9799802E-05
1.672851933	6.729e-01	1.338846022	3.388e-01	3.6579991E-04	0.019748452	9.8590028E-06
0.208763604	7.912e-01	0.602774884	3.972e-01	8.4524055E-04	0.015545017	2.1963527E-05
1.581011562	5.810e-01	1.290906616	2.909e-01	1.0403880E-03	0.005063168	1.0480869E-06
0.75787811	2.421e-01	0.879052822	1.209e-01	3.0401335E-04	0.011015162	2.6494861E-05
1.043712119	4.371e-02	1.021791294	2.179e-02	1.0384492E-03	0.031889041	1.3136252E-05

n=12 时解出各量的误差柱图如下图所示，五个柱分别表示从左至右采用以上各法的结果误差，可见误差随着方法的改进显著减小，在阶数不太大的情况下 Tikhonov 和 CG 都是很好的算法。Gauss 消去法则有显著的误差，在高阶数病态条件下让机器计算的结果几乎没有可信度。

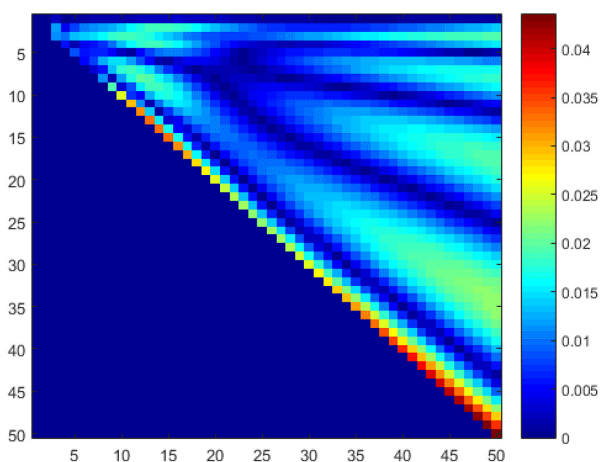


对后三种方法（Tikhonov 正则化方法、SOR 和共轭梯度法）按原有条件计算到 50 阶，绘制出其误差图分别如下（横坐标为阶数，纵坐标 n 为该阶中对应 y_n 的误差），可见共轭梯度法在极度病态的条件下经过较少迭代也能很快地收敛到相当高的精度：

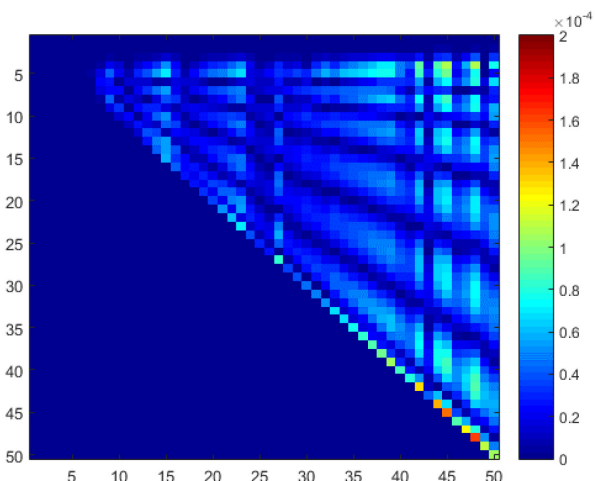
Tikhonov 正则化方法



SOR 迭代法（2000 次迭代）



共轭梯度法（100 次迭代）



五、 结论

高斯消去法虽然有计算简单、容易理解等优点，但是在高阶病态矩阵的机器运算中其误差也是达到显著影响计算结果的地步。由于存在舍入误差和有效位数的限制，对于机器计算需要使用不同的算法。

条件数是衡量矩阵病态特性的指标，具体表现为矩阵离奇异阵的距离。求解高条件数的线性方程组时主要有两大类优化方法，一种是试图降低矩阵条件数来进行直接求解或求出近似解，包括 Cholesky 分解、行/列平衡法和正则化方法如 Tikhonov 正则化方法。另一种是通过对一个不精确解的迭代来逼近精确解，包括 Jacobi 迭代法、Gauss-Seidel 迭代法、超松弛 (SOR) 迭代法，以及通过变分原理演变出来的最速下降法和共轭梯度 (CG) 法。其中，由于性能优异，共轭梯度法常被用来求解各种工程问题如航天器的轨道优化问题。在对精度要求不特别高的情况下，Tikhonov 正则化方法也是运算量比较小的一种方法。

以 Hilbert 矩阵为例，在阶数较低 ($n < 10$) 的情况下，Gauss 消去法和 Cholesky 分解法比起其它三种方法仍具有精度上的优势，然而随着阶数的增加，其误差呈指数式增长，在 12 阶之后开始出现没有有效位数的情况，在 50 阶时误差达到 31096%；相比之下迭代法和正则化方法在高阶数下具有显著的精度优势，是求解大型病态满阵的首选方法。